Thermodynamic graph rewriting

Vincent Danos, Russ Harmer, Ricardo Honorato-Zimmer

Thermodynamic graph-rewriting

Vincent Danos[1], Russ Harmer[2], and Ricardo Honorato-Zimmer

[1] School of Informatics, University of Edinburgh
[2] CNRS & Université Paris-Diderot

Abstract. We develop a new 'thermodynamic' approach to stochastic graph-rewriting. The ingredients are a finite set of reversible graph-rewriting rules $\mathcal{G}$ (called generating rules), a finite set of connected graphs $\mathcal{P}$ (called energy patterns), and an energy cost function $\epsilon : \mathcal{P} \to \mathbb{R}$. The idea is that $\mathcal{G}$ defines the qualitative dynamics by showing which transformations are possible, while $\mathcal{P}$ and ϵ specify the long-term probability π of any graph reachable under $\mathcal{G}$. Given $\mathcal{G}$, $\mathcal{P}$, we construct a finite set of rules $\mathcal{G}_{\mathcal{P}}$ which (i) has the same qualitative transition system as $\mathcal{G}$, and (ii) when equipped with suitable rates, defines a continuous-time Markov chain of which π is the unique fixed point. The construction relies on the use of site graphs and a technique of 'growth policy' for quantitative rule refinement which is of independent interest. The 'division of labour' between the qualitative and the long-term quantitative aspects of the dynamics leads to intuitive and concise descriptions for realistic models (see the example in §4). It also guarantees thermodynamical consistency (*aka* detailed balance), otherwise known to be undecidable, which is important for some applications. Finally, it leads to parsimonious parameterizations of models, again an important point in some applications.

1 Introduction

Along with Petri nets, communicating finite state machines, and process algebras, models of concurrent systems based on graphs and graph transformations (GTS) have long been investigated as means to describe, verify and synthesize distributed systems [11]. Beyond their visual aspect, which is useful in modelling situations, there is a lot to like about GTSs: there are category-theoretic frameworks to express them and encapsulate their syntax; and the existence of a strong meta-theory [19] is a reassurance that methodologies developed in specific cases can be 'ported' to other variants.

Graph-rewriting rules are convenient for writing compact models and modifying them [7], and lend themselves naturally to probabilistic extensions [16,18]. However, for all their flexibility, even rules can only do so much. We ask in this paper "what if we did not have to write the rules?". This is where we take a page from the book of classical statistical mechanics. In such models, which often involve graph-like structures as in the Ising model, the dynamics is not described upfront. Instead, the system of interest is equipped with an 'energy

lansdcape' which specifies its long run behaviour, be it deterministic as in classical mechanics, or probabilistic in statistical physics. The dynamics just follows from the energy data. In the eye of a computer scientist, this use of energy looks like a latent syntax. (This is especially true in the application of these ideas to molecular dynamics.)

The broad aim of this paper is to make this syntax explicit by introducing energy patterns and costs from which the total energy of a state of the system can be computed; and to define a procedure whereby, indeed, the dynamics described as probabilistic graph-rewriting rules can be derived from these energy data. Descriptively, this takes us to an entirely new level of conciseness (as in the example §4). It also guarantees thermodynamical consistency, otherwise known to be undecidable [8], which is important for some applications. But perhaps the nicest byproduct of this approach is the fact that the methodology leads to parsimonious parameterizations. The parameter space which usually scales as the number of rules (which in turn has at best a logarithmic impact on the cost of a simulation event [4]), will now scale as the number of energy patterns provided in the specification.

The particular kind of GTSs we consider forms a reversible subset of the Kappa site-graph stochastic rewriting language. Kappa is used for the simulation and analysis of combinatorial dynamical systems as typically found in cellular signalling networks [20,24] and has been predicted to "become one of the mainstream modelling tools of systems biology within the coming decade" [1]. Similar graph formalisms where nodes have a controlled valence/degree have been considered *e.g.* the BNG language [12,17], Kissinger and Dixon's quantum proof language [10], and Kirchner *et al.* chemical calculi [3]. Site-graph rewriting has found recently a 'home' both in the single-pushout GTS tradition [5] and the double-pushout one [14]. This makes one hopeful that the thermodynamic methodology we propose can crossover to other fields where quantitative GTSs can be used, *e.g.* in the modelling of adaptive networks [13]. While our scalable energy-based parameterization is particularly important in biological applications where parameters often need to be inferred, one can imagine it to be useful in other modelling situations with uncertainty.

Outline: We start with the definition and relevant properties of the specific GTS we use, namely a simple reversible fragment of Kappa. Next, we introduce growth policies (adapted from Ref. [23]), a tool which allows one to replace a rule with an orthogonal set of refined rules while preserving the quantitative semantics. We use this tool with a specific policy which refines a rule into finitely many rules, each of which has a definite energy balance with respect to a given set of energy patterns. This leads to our main theorem which guarantees that the stochastic dynamics of the obtained refined rule set converges to an equilibrium distribution parametrized by the cost of each energy pattern. Throughout, the presentation is set in category-theoretic terms and mostly self-contained. A substantial example concludes the paper. (For lack of space, and following the advice of the referees, proofs were omitted in this extended abstract; these will be presented in a longer version.)

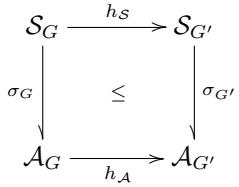
2 Site graph rewriting

2.1 Site graphs and homomorphisms

A *site graph* G consists of a finite sets of *agents* and *sites*, $\mathcal{A}_G$ and $\mathcal{S}_G$, a partial function $\sigma_G : \mathcal{S}_G \rightharpoonup \mathcal{A}_G$, and a symmetric *edge* relation $\mathcal{E}_G$ on $\mathcal{S}_G$. The pair $\mathcal{A}_G$, $\mathcal{E}_G$ form an undirected graph; sites not in the domain of $\mathcal{E}_G$ are said to be *free*. The role of the additional map σ_G is to assign sites to agents; sites not in the domain of σ_G are said to be *dangling*, and will be used to represent half-edges (see below). Usually one also endows agents and/or sites with states (see §4); the construction we will give in §3 carries over trivially to these.

One says G is *realizable* iff (i) no site has an edge to itself; (ii) sites have at most one incident edge; (iii) no dangling site is free; and, (iv) edges have at most one dangling site.

A *homomorphism* $h : G \to G'$ of site graphs is a pair of functions, $h_{\mathcal{S}} : \mathcal{S}_G \to \mathcal{S}_{G'}$ and $h_{\mathcal{A}} : \mathcal{A}_G \to \mathcal{A}_{G'}$, such that (i) whenever $h_{\mathcal{A}}(\sigma_G(s))$ is defined, so is $\sigma_{G'}(h_{\mathcal{S}}(s))$ and they are equal; and (ii) if $s \; \mathcal{E}_G \; s'$ then $h_{\mathcal{S}}(s) \; \mathcal{E}_{G'} \; h_{\mathcal{S}}(s')$.



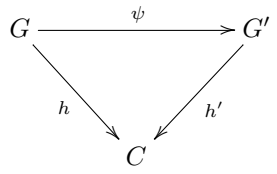
A homomorphism $h : G \to G'$ is an *embedding* iff (i) $h_{\mathcal{A}}$ and $h_{\mathcal{S}}$ are injective; and (ii) if s is free in G, so is $h_{\mathcal{S}}(s)$ in G'. If $h : G \to G'$ is an embedding and G' is realizable then G is also realizable.

Site graphs and homomorphisms form a category **SG** with the natural 'tiered' composition; embeddings form a subcategory; if in addition, we restrict objects to be realizable, we get the subcategory **rSGe** of realizable site graphs and embeddings.

2.2 The category of site graphs over C

A homomorphism $h : G \to C$ is a *contact map* over C iff (i) G is realizable, (ii) σ_C is total and (iii) whenever $h_{\mathcal{S}}(s_1) = h_{\mathcal{S}}(s_2)$ and $\sigma_G(s_1) = \sigma_G(s_2)$, then $s_1 = s_2$. The third condition of local injectivity means that every agent of G has at most one copy of each site of its corresponding agent in C; C is called the *contact graph*.

Hereafter, we work in the (comma) category $\mathbf{rSGe}_C$ whose objects are contact maps over C, and arrows are embeddings such that the associated triangle commute in **SG**. We write $\Upsilon(h, h')$ for the set of such embeddings between h, h' contact maps over C; we also write $|\_|$ for the domain functor from $\mathbf{rSGe}_C$ to **rSGe** which forgets types. In particular, if $h : G \to C$ is a contact map, we write $|h|$ for its source G.



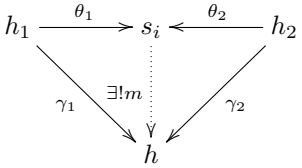
The contact graph C is fixed and plays the role of a *type*: it specifies the kinds of agents that exist, the sites that they may possess, and which of the

$|\mathcal{S}_C|^2$ possible edge types are actually valid. It also gives canonical names to the types of agents and their sites. In examples, we write agent and sites names directly.

In **rSGe**, a dangling site s in G can be mapped to any site of G' by an embedding $h : G \to G'$; as such, it can be used as an *any site* wild card when matching G. In $\mathbf{rSGe}_C$, the contact map $c : G \to C$ tells us which agent A in C the site s belongs to because σ_C is total, and this must be respected by h; we call this a *binding type* wild card since it typically allows us to express the property of being bound to the site s of *some occurrence* of the agent A.

The category **SG** has all pull-backs, constructed from those in **Set**; it is easy to see that they restrict to $\mathbf{rSGe}_C$. The category **SG** also has sums, but these do not restrict to $\mathbf{rSGe}_C$. (Just like sums in **Set** do not restrict to the subcategory of injective maps.)

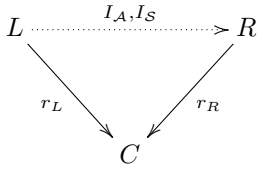
However, $\mathbf{rSGe}_C$ has *multi-sums*: meaning for all pairs of site graphs of type C, $h_1 : G_1 \to C$ and $h_2 : G_2 \to C$, there exists a family of co-spans $\theta_1^i : h_1 \to s_i \leftarrow h_2 : \theta_2^i$, such that any co-span $\gamma_1 : h_1 \to h \leftarrow h_2 : \gamma_2$ factors through *exactly one* of the family and does so *uniquely*. The idea is that the pairs θ_1^i, θ_2^i enumerate all minimal ways in which one can glue h_1 and h_2, that is to say all the minimal glueings of G_1 and G_2 that respect C. There are *finitely* many which all factor through the standard sum in the larger slice category $\mathbf{SG}_C$.



The notion of multi-sum dates back to Ref. [9]; we will call them *minimal glueings* in **rSGe** according to their intuition in this concrete context, and use them in §3.2 to construct balanced rules.

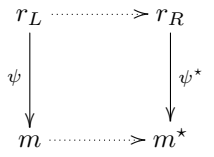
2.3 Rules

A *rule* r over C is a pair of contact maps $r_L : L \to C$, $r_R : R \to C$ which differ only in their edge structures, *i.e.* $\mathcal{A}_L = \mathcal{A}_R$, $\mathcal{S}_L = \mathcal{S}_R$, $\sigma_L = \sigma_R$, $r_{L\mathcal{A}} = r_{R\mathcal{A}}$, and $r_{L\mathcal{S}} = r_{R\mathcal{S}}$.



A contact map $m : M \to C$ is a *mixture* iff σ_M is total (no dangling edge) and, for all $a \in \mathcal{A}_M$, $\sigma_M^{-1}(a) \cong \sigma_C^{-1}(m_{\mathcal{A}}(a))$, *i.e.* $m_{\mathcal{S}}$ is locally surjective. In words, a mixture is a *fully-specified* site graph with respect to the type C.

An embedding $\psi : r_L \to m$ induces a *rewrite* of a mixture m by modifying the edge structure of the image of ψ, *i.e.* an instance of L in M, to that of R; the result of rewriting is a new mixture $m^\star$, where $|m^\star|$ has the same agents and sites as $M = |m|$, and an embedding $\psi^\star : r_R \to m^\star$. This can be formalized using double push-out rewriting [5] (since all the required push-outs do exist in $\mathbf{rSGe}_C$). But with the simple rules considered here, there is no need.



(1)

We also write $\Upsilon(r, m)$ for the set of all embeddings $\psi : r_L \to m$.

The inverse of r, defined as $r^\star := (r_R, r_L)$ is also a valid rule; by rewriting $m^\star$ with $r^\star$ via $\psi^\star$, we recover m and ψ.

Given a finite set of rules $\mathcal{G}$ over C, we define a labelled transition system $\mathcal{L}_\mathcal{G}$ on mixtures over C: a transition from a mixture m is a rewriting step determined and labelled by an 'event' (r, ψ) as in diagram (1); with r in $\mathcal{G}$, and ψ in $\Upsilon(r, m)$.

We suppose hereafter that $\mathcal{G}$ is closed under rule inversion, *i.e.* $\mathcal{G} = \mathcal{G}^\star$. Hence, every (r, ψ)-transition has an inverse $(r^\star, \psi^\star)$, and $\mathcal{L}_\mathcal{G}$ is symmetric.

2.4 CTMC semantics

It is not difficult to see that for any rule r, $|\Upsilon(r, m)| \leq |\mathcal{A}_{|m|}|^{d(r)}$ where $d(r)$ is the number of connected components in r_L. Hence, $\mathcal{L}_\mathcal{G}$ has finite out-degree, bounded by $|\mathcal{G}| \cdot |\mathcal{A}_{|m|}|^d$ for some d. Also, as agents are preserved by rules, the (strongly) connected components of $\mathcal{L}_\mathcal{G}$ are finite.

Hence, given a *rate map* k from $\mathcal{G}$ to $\mathbb{R}_+$, we can equip $\mathcal{L}_\mathcal{G}$ with the structure of an irreducible continuous-time Markov chain (CTMC), simply by assigning rate $k(r)$ to an event of the form (r, ψ).

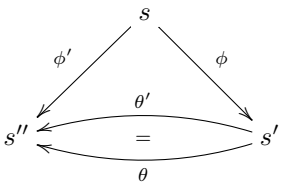
We write $\mathcal{L}^k_\mathcal{G}$ for the obtained CTMC.

We need here to record a definition for later use: a finite CTMC $\mathcal{M}$ has *detailed balance* for a probability distribution π on $\mathcal{M}$'s state space, if for all states x and y, $\pi(x) \cdot q(x, y) = \pi(y) \cdot q(y, x)$ where $q(x, y)$ is $\mathcal{M}$'s transition rate from x to y. This implies that, assuming $\mathcal{M}$ is irreducible, π is the unique fixed point of the action of $\mathcal{M}$, to which the probabilistic state of $\mathcal{M}$ converge, regardless of the initial state.

2.5 Extensions and rule refinement

Epis of $\mathbf{rSGe}_C$ can be characterized as follows [23]: suppose $s : G \to C$ and $s' : G' \to C$ are contact maps then $\phi : s \to s'$ of $\mathbf{rSGe}_C$ is an epi iff every connected component of G' contains at least one agent in the image of $\phi_\mathcal{A}$.

We refer to an epi $\phi : s \to s'$ as an *extension* of s. The category of extensions of s is a pre-order, *i.e.* there is at most one arrow between any two objects: if $\phi' = \theta\phi = \theta'\phi$ then $\theta = \theta'$ because ϕ is an epi. We write $\phi \leq \phi'$ for this *specialization* order. If $\phi \leq \phi'$ and $\phi' \leq \phi$ then we write $\phi \cong_s \phi'$.



(2)

A family of epis $\phi_i : s \to t_i$ *uniquely decomposes* s iff, for all mixtures m and embeddings $h : s \to m$, there exists a unique i and ψ such that $h = \psi\phi_i$. This is the basic requirement for a reasonable notion of rule refinement: it guarantees that the LHS s of a given rule splits into a non-overlapping collection of more specific cases t_i.

In the next Section, we will be constructing specific such decompositions in order to produce families of sub-rules which are compatible with energy patterns.

First, we recall the growth policy method to find such unique decompositions which works by detailing which agents and sites should be added to s.

Specifically, a *growth policy* Γ for s is a family of functions Γ_ϕ, indexed by extensions $\phi : s \to t$, where Γ_ϕ maps $u \in \mathcal{A}_{|t|}$ to a subset $\Gamma_\phi(u)$ of $\sigma_C^{-1}(t_\mathcal{A}(u))$, *i.e.* each agent in $|t|$ is allocated a subset of the sites its sites can map to in C. An agent in $|t|$ may cover some, or all, of these sites or even completely extraneous sites: if the former, *i.e.* for all u in $\mathcal{A}_{|t|}$, $t_\mathcal{S}(\sigma_{|t|}^{-1}(u)) \subseteq \Gamma_\phi(u)$, we say that ϕ is *immature*; if for all us, the inclusion is an equality, we say that ϕ is *mature*; otherwise ϕ is said to be *overgrown*. The functions Γ_ϕ must satisfy, for all extensions ϕ and $\phi' \geq \phi$, the *faithfulness* property, $\Gamma_\phi = \Gamma_{\phi'}\psi_\mathcal{A}$, where ψ is the epi witnessing $\phi \leq \phi'$; so a site requested by ϕ must be requested by any further extension. If ϕ is not overgrown then no $\phi' \leq \phi$ is overgrown either. Also, note that the *union* of two growth policies is itself a growth policy.

Given an s and a growth policy Γ for s, we define $\Gamma(s)$ by choosing one representative per $\cong_s$-isomorphism class of the set of all extensions of s which are mature according to Γ.

Theorem 1 *If Γ is a growth policy for s, then $\Gamma(s)$ uniquely decomposes s.*

The theorem (adjusted from Ref. [23]) guarantees that factorizations through $\Gamma(s)$ are unique, but not that they always exist. In the next section, we will construct a growth policy for which this propoerty of exhaustivity of the decomposition can be proved by hand.

Given a rule r and an extension ϕ of r_L, r_ϕ denotes the 'refined' rule associated to ϕ. If Γ is a growth policy for r_L, the *refinement* of r by Γ is the set of rules, $\Gamma(r)$, the elements of which are of the form r_ϕ, for ϕ in $\Gamma(r_L)$ a mature extension.

It is easy to see that due to the simple nature of our rules, the category of extensions of r_L and r_R are isomorphic; if ϕ is an extension of r_L, we will write $\phi^\star$ for the corresponding extension of r_R.

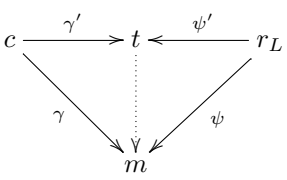
An example of growth policy is the *ground* policy which assigns all possible sites to all agents. In which case: $\Gamma(s)$ is simply the set, possibly infinite, of epis of s into mixtures, considered up to $\cong_s$; and $\Gamma(r)$, the ground refinement of r, contains all refinements of r along these epis, which therefore directly manipulate mixtures.

3 Rule generation

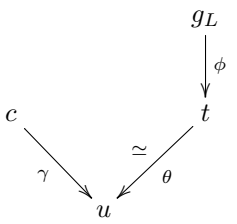
We fix a finite set $\mathcal{G}$ of *generator* rules; and a finite set $\mathcal{P}$ of connected contact maps in $\mathbf{rSGe}_C$; these are our *energy patterns*.

The goal is now to refine $\mathcal{G}$ into a new rule set $\mathcal{G}_\mathcal{P}$ where each refined rule is $\mathcal{P}$-*balanced*, which means that, however applied, it consumes or produces a fixed amount of each pattern c in $\mathcal{P}$. The construction proceeds in two steps: first, we characterize balanced refinements; second, we define a growth policy with balanced mature extensions, and apply Th. 1. Note that ground extensions of g are trivially balanced but, in general, the ground refinement is impractically large or even infinite; ours will always be finite.

3.1 $\mathcal{P}$-balanced extensions



Consider c in $\mathcal{P}$, and a rule r. For an r-event ψ to *consume* an instance γ of c in a mixture m, the cospan $(\gamma_{\mathcal{S}}, \psi_{\mathcal{S}})$ must have images which intersect on at least one edge modified by r. This is the case iff the associated minimal glueing (γ', ψ') —obtained by restricting the cospan to the union of its images in m— has the same property. Likewise, for an r-event to *produce* an instance of c, the associated minimal glueing between c and r_R must have a modified intersection. We call such minimal glueings *relevant*; they are the ones which underlie events that can affect the set of instances of c.



(3)

Pick g in $\mathcal{G}$ and $\phi : g_L \to t$ an extension of g_L. One says that ϕ is $\mathcal{P}$-*left-balanced* iff, for all relevant minimal glueings $\gamma : c \to u \leftarrow t : \theta$ with $c \in \mathcal{P}$, θ is an isomorphism. This means that the image of c under γ is contained in t. Symmetrically, one says that ϕ is $\mathcal{P}$-*right-balanced* iff $\phi^\star$ is a $\mathcal{P}$-left-balanced extension of $r^\star$.

An extension ϕ is $\mathcal{P}$-*balanced* iff it is $\mathcal{P}$-left- and $\mathcal{P}$-right-balanced; we say that ϕ is *prime* iff it is minimal $\mathcal{P}$-balanced in the specialization order $\leq$.

If ϕ is a $\mathcal{P}$-balanced extension of g, the refined rule g_ϕ has a *balance vector* in $\mathbb{Z}^{\mathcal{P}}$, written $\Delta\phi$, where $\Delta\phi(c)$, for $c \in \mathcal{P}$, is the amount of c produced by *any* g_ϕ-event leading from m to $m^\star$, or equivalently the difference between the number of embeddings of c in the RHS and the LHS of g_ϕ. Indeed, as ϕ is balanced, $|\,\Upsilon(c, m^\star)| - |\,\Upsilon(c, m)| = |\,\Upsilon(c, g_{\phi,R})| - |\,\Upsilon(c, g_{\phi,L})|$.

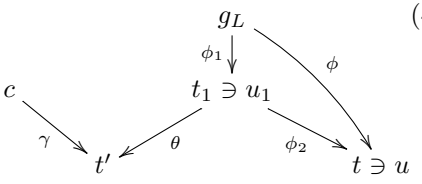
Conversely, a non-$\mathcal{P}$-balanced extension will incur different $\Delta\phi(c)$ for well-chosen applications of g_ϕ, if c in $\mathcal{P}$ violates the condition of diagram (3). Thus, the notion of balanced extension characterizes the property that we want. (This would no longer be the case if one were to relativize the construction to a superset of reachables; *e.g.* in order to reduce the size of the generated rule set.)

3.2 Add-by-glueing

We now define a growth policy, which uses minimal glueings on non-$\mathcal{P}$-balanced extensions ϕ to add further required sites into ϕ's codomain; this corresponds in diagram (3) to the case where θ is not an isomorphism.

Some care is needed to ensure faithfulness, *i.e.* $\Gamma_\phi = \Gamma_{\phi'\phi}\,\phi'_{\mathcal{A}}$, since relevant minimal glueings on ϕ can disappear along a further extension ϕ' and, consequently, a site that was 'requested' at ϕ may no longer be so after at $\phi'\phi$. To address this, we add site requests from all relevant minimal glueings in the *past* of an extension.

Given $g \in \mathcal{G}$ we define a growth policy Γ_L for g_L. Suppose $\phi : g_L \to t$ is an extension of g_L. We set $\Gamma_L(\phi)$ to request a site s in $\sigma_C^{-1}(t_{\mathcal{A}}(u))$ at agent $u \in \mathcal{A}_{|t|}$ iff either (i) $u = \phi_{\mathcal{A}}(u_0)$ and $s = \phi_{\mathcal{S}}(s_0)$ for some u_0 in $\mathcal{A}_{|g_L|}$, s_0 in $\mathcal{S}_{|g_L|}$; or (ii) ϕ factorizes as $\phi_2\phi_1$, where $\phi_1 : s \to t_1$, and there is a relevant minimal glueing $\gamma : c \to t' \leftarrow t_1 : \theta$, u_1 in $\mathcal{A}_{|t_1|}$, and a site s'_1 in $\sigma^{-1}_{|t'|}(\theta_{\mathcal{A}}(u_1))$ such that $u = \phi_{2,\mathcal{A}}(u_1)$, and $s = t'_{\mathcal{S}}(s'_1)$.



(4)

The first clause simply ensures that all sites already covered in g_L are asked for; the second one adds in sites which appear by glueing at some point between g_L and t. We refer to $\phi_2 : t_1 \to t$ as a *rewind* of ϕ.

Symmetrically, we define a growth policy Γ_R for g_R by applying the same definition to the reverse generator $g^\star$. Since extensions of g_L and g_R are isomorphic, we can, with a slight abuse of notation, define $\Gamma^{\mathcal{P}} := \Gamma_L \cup \Gamma_R$.

Theorem 2 *The above $\Gamma^{\mathcal{P}}$ is indeed a growth policy for g_L; the induced refined rule set $\Gamma^{\mathcal{P}}(g)$ is non-empty, balanced, exhaustive and finite.*

Therefore, given $\mathcal{G}$ and $\mathcal{P}$, we obtain a finite $\mathcal{P}$-balanced rule set which refines $\mathcal{G}$ exhaustively, by setting $\mathcal{G}_{\mathcal{P}} := \dot{\cup}_{g \in \mathcal{G}} \Gamma^{\mathcal{P}}(g)$ (disjoint sum). To every refinement g_ϕ, corresponds an inverse refinement $g^\star_{\phi^\star}$; hence, $\mathcal{G}_{\mathcal{P}} = \mathcal{G}^\star_{\mathcal{P}}$ is closed under inversion like $\mathcal{G}$.

3.3 Rates

To equip $\mathcal{G}_{\mathcal{P}}$ with rates, we suppose given a $\mathcal{P}$-indexed real-valued vector of *energy costs* ϵ, and a rate map $k : \mathcal{G}_{\mathcal{P}} \to \mathbb{R}_+$ such that, for all g_ϕ in $\mathcal{G}_{\mathcal{P}}$:

$$\log k(g^\star_{\phi^\star}) - \log k(g_\phi) = \epsilon \cdot \Delta\phi \tag{5}$$

with $\Delta\phi$ in $\mathbb{Z}^{\mathcal{P}}$, the balance vector of the refined rule g_ϕ with respect to $\mathcal{P}$, a well-defined quantity by Th. 2.

We write $\mathcal{P}(x)$ for the $\mathcal{P}$-indexed vector which maps c to $|\Upsilon(c, x)|$, and define the *energy* $E(x)$ of x as $\epsilon \cdot \mathcal{P}(x)$. We also write $\mathcal{L}_{\mathcal{G}}(x)$ for the finite (strongly) connected component of x in $\mathcal{L}_{\mathcal{G}}$, and define a probability distribution (in Boltzmann format) on $\mathcal{L}_{\mathcal{G}}(x)$ by:

$$\pi_x(y) := e^{-\epsilon \cdot \mathcal{P}(y)} / \sum_{y \in \mathcal{L}_{\mathcal{G}}(x)} e^{-\epsilon \cdot \mathcal{P}(y)} \tag{6}$$

Theorem 3 *Let $\mathcal{G}$, $\mathcal{P}$, $\mathcal{G}_{\mathcal{P}}$, k, and π_x be as above; $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}$ and $\mathcal{L}_{\mathcal{G}}$ are isomorphic as symmetric LTSs; and, for any mixture x, the irreducible continuous-time Markov chain $\mathcal{L}^k_{\mathcal{G}_{\mathcal{P}}}$ has detailed balance for, and converges to π_x, on $\mathcal{L}_{\mathcal{G}_{\mathcal{P}}}(x) = \mathcal{L}_{\mathcal{G}}(x)$ the finite strongly connected component of x.*

Note that the subset of the state space which is reachable from x in $\mathcal{L}_\mathcal{G}$, namely $\mathcal{L}_\mathcal{G}(x)$ is finite; hence, the *partition function* $Z(x) = \sum_{y \in \mathcal{L}_\mathcal{G}(x)} e^{-E(y)}$ is finite. With rules which increase the number of agents, components $\mathcal{L}_\mathcal{G}(x)$ can be infinite, and $Z(x)$ may diverge. For (mass action stochastic) Petri nets, convergence is guaranteed if detailed balance holds, but it is not true in general for Kappa [8,6].

Another point worth making is that the result holds symbolically—regardless of the energy cost ϵ. So ϵ can be seen as a set of parameters, an ideal support for machine learning techniques if one were contemplating fitting a model to data.

3.4 A linear kinetic model

So, the reader will ask, what of the actual rates of $\mathcal{L}^k_{\mathcal{G}_\mathcal{P}}$? Among all possible choices which accord with (5), it is possible to delineate a tractable subset the size of which grows quadratically in $|\mathcal{P}|$. This is a useful log-linear heuristics, which is common in machine learning but has no claim to validity.

We keep the same notations as in Th. 3.

Suppose we have, for every generating rule g in $\mathcal{G}$, a constant $c_g \in \mathbb{R}$, and a matrix A_g of dimension $|\mathcal{P}| \times |\mathcal{P}|$. Subject to the constraints that $c_{g^\star} = c_g$, and $A_{g^\star} + A_g = I$, we can define a log-affine rate map which satisfies (5) by:

$$\log(k(g_\phi)) := c_g - A_g(\epsilon) \cdot \Delta\phi \tag{7}$$

The kinetic model expressed in (7) requires of the order of $|\mathcal{P}|^2 \times |\mathcal{G}|$ parameters. In practice, one needs even fewer parameters, as only those energy patterns that are relevant to a given g, *i.e.* have a non-zero balance for at least one rule in $\Gamma^\mathcal{P}(g)$, need to be considered when building A_g. Typically, for larger models, this will be a far smaller number than $|\mathcal{P}|$. This relative parsimony is compounded by the fact that the number of *independent* parameters will be often lower, because the $\Delta\phi$ family has often low rank. It is to be compared with the total number of choices possible which is far greater as it is of the order of the number of refinements, that is to say $\sum_{g \in \mathcal{G}} |\Gamma^\mathcal{P}(g)|$.

If we set $c_{g^\star} = c_g = 0$, $A_{g^\star} = 0$, $A_g = I$, we get: $k(g_\phi) = e^{-\epsilon \cdot \Delta\phi}$, $k(g^\star_{\phi^\star}) = 1$. As $\epsilon \cdot \Delta\phi$ is the difference of energy between the target and source in any application g_ϕ, this choice amounts to being exponentially reluctant to climb up the energy gradient. This is a continuous-time version of the celebrated Metropolis algorithm [22].

4 Allosteric ring

We can put our energy-based modelling methodology to use on a realistic example of a bacterial flagellar engine. In this section, we will prefer the traditional syntax of Kappa to denote site graphs: namely subscripts for states and shared superscripts for edges between sites, *e.g.* $A(x^1_0), B(y^1)$. Differently from the mathematical definitions of §2, agent and site types are indicated as explicit labels.

We use KaSim (`https://github.com/jkrivine/KaSim`), the standard Kappa engine, for the simulation shown below.

The engine can rotate clockwise or anti-clockwise at high angular velocities, and this will decide whether the bacterium tumbles or swims forward. One can build a simple model of the switch between the two modes [2]. The engine is seen as a ring of n identical components, P, with two possible conformations, 0 and 1. (In reality, each of the $n = 34$ component protomers is itself a tiny complex made of different subcomponents, but the model ignores this.) A ring homogeneously in state 0 (1) rotates (anti-) clockwise and induces tumbling (straight motion). Importantly, neighbouring Ps on the ring prefer to have matching conformations. States of the ring with many mismatches thus incur high penalties. In the absence of any Y molecule binding a P, its favoured conformation is 0; conversely, in the presence of a Y, P favours 1. (Y stands for a small diffusible protein named *CheY*.) To bind, Y has to be activated by an external signal. Hence the switch can be triggered by a sudden activation of Y which then binds the ring and induces a change of regime. The sharper the transition between the two regimes the better.

As each of the Ps can be in four states, the ring has on the order of 10^{20} non-isomorphic configurations which precludes any reaction-based (*e.g.* Petri nets) approach to the dynamics where each global state is considered as one chemical species. At this stage, we could apply the rule-based approach, or, better, we can obtain the rules *indirectly* by applying the methodology of §3. This is what we do now informally.

First, we define our contact graph with two agent types: $P(x, y, f_{0,1}, s)$ with domains x, y to form the ring, s to bind its signal Y, and f a placeholder for P's conformation; $Y(s_{u,p})$ with two internal states to represent activity.

Second, we capture the informal statements in the discussion above by defining the energy patterns and associated costs. Note that the various motifs overlap. Following §3, we associate to each ring configuration x the occurrence vector $\mathcal{P}(x)$ and total energy $\epsilon \cdot \mathcal{P}(x)$. For example, a ring of size n uniformly in state 0 and with no bound Ys has total energy $n(\epsilon^{PP}_{00} + \epsilon^{P}_{0})$. This, in turn, defines the equilibrium distribution of the ring, namely x has probability proportional to $\exp(-\epsilon \cdot \mathcal{P}(x))$. (The convention is that the lower the energy, the likelier the state.)

Motif	*Cost*
$P(f_i, x^1), P(y^1, f_j)$	ϵ^{PP}_{ij}
$P(f_i)$	ϵ^{P}_{i}
$P(f_i, s^1), Y(s^1)$	ϵ^{PY}_{i}

In order to complete our energy landscape, we need to pick energy costs which reward or penalize local configurations as discussed above: the role of (8) is to align the internal states of neighbours on the ring — an Ising penalty term for mismatching neighbours which will "spread conformation"; (9) makes 0 the favoured state, while (10), which kicks in only in the presence of Y, makes 1 the favoured state.

$$\epsilon^{PP}_{00}, \epsilon^{PP}_{11} < \epsilon^{PP}_{10}, \epsilon^{PP}_{01} \quad (8)$$

$$\epsilon^{P}_{0} < \epsilon^{P}_{1} \quad (9)$$

$$\epsilon^{PY}_{0} > \epsilon^{PY}_{1} \quad (10)$$

The next step is to create the dynamics. The naive rule b for PY binding:

$$b := P(s), Y(s_p) \leftrightarrow P(s^1), Y(s_p^1)$$

has a ΔE which is ambiguous as it will be either ϵ_0^{PY} or ϵ_1^{PY} depending on its instances; hence, we have no hope of assigning rates to this rule that satisfy detailed balance—unless $\epsilon_0^{PY} = \epsilon_1^{PY}$, which contradicts (10). To get a definite balance, one needs to refine this rule:

$$b_0 := P(f_0, s), Y(s_p) \leftrightarrow P(f_0, s^1), Y(s_p^1)$$
$$b_1 := P(f_1, s), Y(s_p) \leftrightarrow P(f_1, s^1), Y(s_p^1)$$

Now each rule b_i specifies enough of the context in which it applies to have a definite energy balance ϵ_i^{PY}. Following the same intuition of revealing (just) enough context, we obtain a balanced rule set for conformational changes:

$$f_{ij} := P(f_i, y^1), P(x^1, f_0, y^2, s), P(x^2, f_j) \leftrightarrow P(f_i, y^1), P(x^1, f_1, y^2, s), P(x^2, f_j)$$
$$f'_{ij} := P(f_i, y^1), P(x^1, f_0, y^2, s^-), P(x^2, f_j) \leftrightarrow P(f_i, y^1), P(x^1, f_1, y^2, s^-), P(x^2, f_j)$$

The first (second) group of rules represents the changes in the absence (presence) of a Y bound to the middle P undergoing a change of conformation. (The fact that P's site s is bound is indicated by the underscore exponent.)

These f-rules have respective balance:

$$\epsilon_{i1}^{PP} + \epsilon_{1j}^{PP} - \epsilon_{i0}^{PP} - \epsilon_{0j}^{PP} + \epsilon_1^P - \epsilon_0^P$$
$$\epsilon_{i1}^{PP} + \epsilon_{1j}^{PP} - \epsilon_{i0}^{PP} - \epsilon_{0j}^{PP} + \epsilon_1^P - \epsilon_0^P + \epsilon_1^{PY} - \epsilon_0^{PY}$$

As we have ten reversible rules, and only eight energy patterns, there must be linear dependencies between the various balances. Indeed, in this case, it is easy to see that the family of vector balances has rank six. Thermodynamic consistency induces relationships between rates; a well-established fact in the case of reaction networks (*e.g.* see Ref. [6]).

With the rules in place, the final step is to choose rates which satisfy detailed balance. This guarantees that the obtained rule set converges to the equilibrium specified by the choice of the energy cost vector. Convergence will happen whatever ϵ is, ie symbolically. If, in addition, ϵ follows (8–10), one can see in Fig. 1 that the ring 1) undergoes sharp transitions when active Y is stepped up and down again, and 2) has at all times very few mismatches.

4.1 How to generate the rules

Our set of balanced rules for the ring dynamics was based on two generators, b for binding, f for flipping:

$$b := P(s), Y(s_p) \leftrightarrow P(s^1), Y(s_p^1)$$
$$f := P(f_0) \leftrightarrow P(f_1)$$

Note that there is a design choice here. In effect, we are saying that we are not interested in forming/breaking the bonds between the Ps in the ring. If we

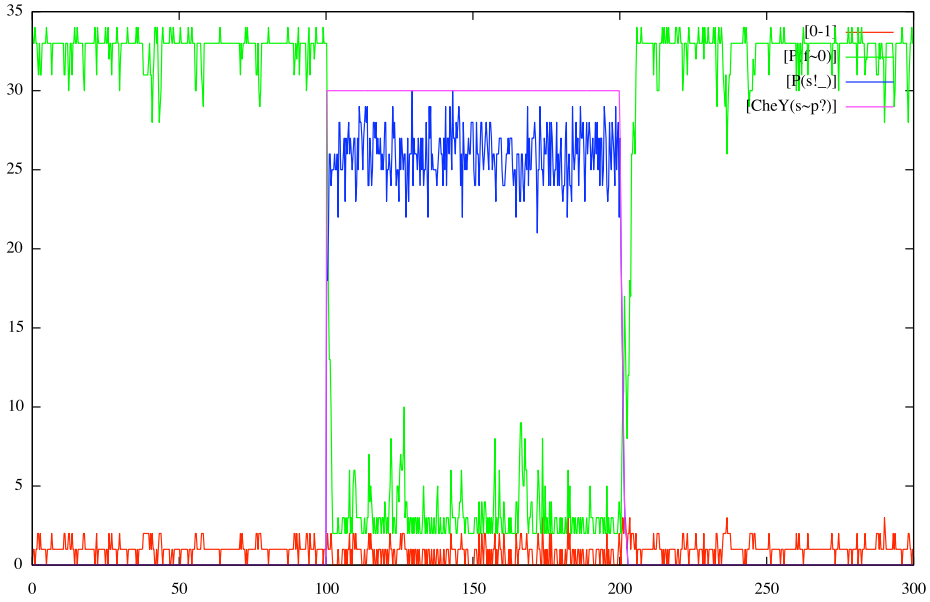


Fig. 1. The simulation steps up the amount of active Y at $t = 100$, and down again at $t = 200$; this sends the entire ring into an homogeneously 1 conformation, and back to 0. The number of mismatches (lowest curve) stays low, even during transitions.

wanted to incorporate also the ring assembly in the model, we would have to add $P(x), P(y) \leftrightarrow P(x^1), P(y^1)$ among our generator set $\mathcal{G}$. This would generate many more refined rules, as we will see. Recall that our patterns fall in three subgroups: $P(f_i, x^1), P(y^1, f_j)$; $P(f_i)$; and $P(f_i, s^1), Y(s^1)$.

Consider the extensions of b: clearly only the last pattern can glue relevantly on it; the corresponding (unique) site request is for P to reveal f and its internal state. This gives the first two rules b_0, b_1.

Consider now the more interesting extensions of f: the second pattern type glues relevantly but does not generate any site request; the third one asks P to reveal its site s, resulting in two possible extensions (s^- means that s is bound):

$$P(f_0, s) \leftrightarrow P(f_1, s)$$
$$P(f_0, s^-) \leftrightarrow P(f_1, s^-)$$

These extensions are *not* mature yet, as one can glue relevantly patterns of the first type on both sides of P, inducing a further request for revealing P's sites x and y. If we are in the component of an initial state where Ps are arranged in a ring, then we know that the neighbours on both sides exist and are Ps; this gives the final refinement of the above into the rules f_{ij}, f'_{ij} described earlier. If, on the other hand, we do not know that, we also have to add several rules where one or both of x, y are free, corresponding to open P-chains. This demonstrates the sensitivity of the obtained rule set to the initial choice of generators.

Hence, the rule set above does accord with our general refinement strategy of §3.

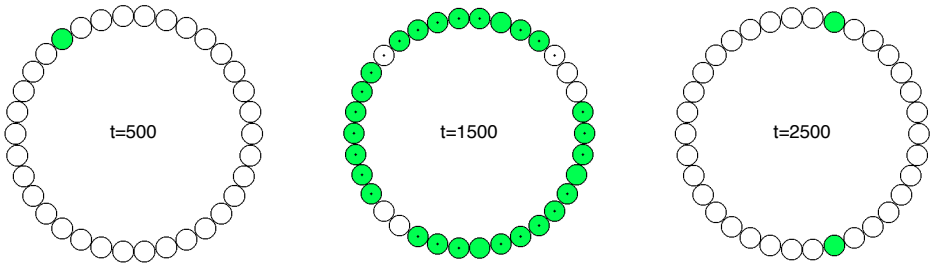


Fig. 2. Snapshots of the ring configuration are taken at time 500, 1500, and 2500. Solid (green) circles indicate conformation 1, hollow ones conformation 0; a dot in the centre indicates a bound (hence active) Y. At times 500, 2500, no Y is bound (because they are all inactive) and the ring is globally in state 0, up to tiny fluctuations; at time 1500, it is globally in state 1 as a consequence of the binding of Ys.

We can visualize the obtained simulations by extracting snapshots before, after and during the injection of active Ys, as in Fig. 2. Again we see few mismatches in both regime because of the Ising interaction expressed by the ϵ^{PP} energy costs. The full model is available on-line at `http://www.rulebase.org/models/ising-ring`. The choice of rates made in Ref. [2] for the f-generator is a particular symmetric case of our model (7), namely $A_{f^\star} = A_f = I/2$.

5 Conclusion

We have presented a new 'energy-oriented' methodology for the development of site graph rewriting models based on a set $\mathcal{P}$ of energy patterns; these patterns use a graphical syntax which allows us to specify the energy landscape. Rewrite rules are implicitly defined by $\mathcal{P}$ and generator rules $\mathcal{G}$. The resulting rule set $\mathcal{G}_\mathcal{P}$ is guaranteed to be thermodynamically correct and to eventually converge to the probability distribution described by the energy landscape given suitable rates. The construction is entirely parametric in the energy costs ϵ, and modular in $\mathcal{G}$. This means that in a modelling context, one can sweep over various values for ϵ without having to rebuild the model, and compositionally add new rule components to $\mathcal{G}$. Both features are clearly useful.

We expect our construction to provide a broad and uniform language to describe and analyse models of interacting biomolecules and similar systems of a quantitative fine-grained and distributed nature.

There are no specific conditions bearing on this construction other than that energy patterns should be local. It would be interesting to investigate whether, suitable constraints on patterns and generator rules can obtain optimized generated rule sets. Another interesting extension would be to deal with non-local

forms of energies expressing long-range interactions, where the metric is read off the graph itself. In practice, there will be many more rules generated, and beyond the descriptive aspects, simulations will need new ideas to be feasible. A ray of hope comes from the log-affine kinetic model (presented in the last subsection), as rules can be partitioned by energy balances for faster selection.

Finally, as said in the introduction, there is a growing body of literature which turns a theoretical eye to site graph rewriting [14,10,15,5], and it is tempting to ask whether our derivation can be replayed in more abstract settings; in particular, it would be very interesting to investigate its integration with the abstract framework for rule-based modelling developed in [21].

Acknowledgments We would like to thank Peter Swain, Andrea Weisse, Julien Ollivier, Nicolas Oury, Finlo Boyde and Eric Deeds for many useful and fascinating conversations concerning the subject of this paper.

References

1. John A Bachman and Peter Sorger. New approaches to modeling complex biochemistry. *Nature methods*, 8(2):130, 2011.
2. F. Bai, R.W. Branch, D.V. Nicolau Jr, T. Pilizota, B.C. Steel, P.K. Maini, and R.M. Berry. Conformational spread as a mechanism for cooperativity in the bacterial flagellar switch. *Science*, 327(5966):685–689, 2010.
3. Olivier Bournez, Guy-Marie Côme, Valérie Conraud, Hélène Kirchner, and Liliana Ibanescu. A rule-based approach for automated generation of kinetic chemical mechanisms. In Robert Nieuwenhuis, editor, *RTA*, volume 2706 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2003.
4. V. Danos, J. Feret, W. Fontana, and J. Krivine. Scalable simulation of cellular signaling networks. *Asian Symposium on Programming Languages and Systems*, pages 139–157, 2007.
5. V. Danos, R. Harmer, and G. Winskel. Constraining rule-based dynamics with types. *Mathematical Structures in Computer Science*, 23(2):272–289, 2013.
6. V. Danos and N. Oury. Equilibrium and termination II: the case of Petri Nets. *Mathematical Structures in Computer Science*, 23(2):290–307, 2013.
7. Vincent Danos. Agile modelling of cellular signalling. *SOS'08 Invited paper, Electronic Notes in Theoretical Computer Science*, 229(4):3–10, 2009.
8. Vincent Danos and Nicolas Oury. Equilibrium and termination. In S. Barry Cooper, Prakash Panangaden, and Elham Kashefi, editors, *Proceedings Sixth Workshop on Developments in Computational Models: Causality, Computation, and Physics*, volume 26 of *EPTCS*, pages 75–84, 2010.
9. Y. Diers. Familles universelles de morphismes. *Tech. report, Université des Sciences et Techniques de Lille I*, 1978.
10. L. Dixon and A. Kissinger. Open graphs and monoidal theories. *arXiv:1011.4114*, 2010.
11. Hartmut Ehrig. *Handbook of graph grammars and computing by graph transformation: Applications, Languages and Tools*, volume 2. World Scientific Publishing Company, 1999.
12. J.R. Faeder, M.L. Blinov, and W.S. Hlavacek. Rule-based modeling of biochemical systems with BioNetGen. *Methods Mol. Biol*, 500:113–167, 2009.

13. Thilo Gross and Hiroki Sayama. *Adaptive networks*. Springer, 2009.
14. Jonathan Hayman and Tobias Heindel. Pattern graphs and rule-based models: The semantics of kappa. In Frank Pfenning, editor, *FoSSaCS*, volume 7794 of *Lecture Notes in Computer Science*, pages 1–16. Springer, 2013.
15. R. Heckel. DPO transformation with open maps. *Graph Transformations*, pages 203–217, 2012.
16. Reiko Heckel. Dpo transformation with open maps. In Hartmut Ehrig, Gregor Engels, Hans-Jörg Kreowski, and Grzegorz Rozenberg, editors, *ICGT*, volume 7562 of *Lecture Notes in Computer Science*, pages 203–217. Springer, 2012.
17. W.S. Hlavacek, J.R. Faeder, M.L. Blinov, R.G. Posner, M. Hucka, and W. Fontana. Rules for modeling signal-transduction systems. *Science Signalling*, 2006(344), 2006.
18. Jean Krivine, Robin Milner, and Angelo Troina. Stochastic bigraphs. *Electronic Notes in Theoretical Computer Science*, 218:73–96, 2008.
19. Stephen Lack and Pawel Sobocinski. Adhesive categories. In Igor Walukiewicz, editor, *FoSSaCS*, volume 2987 of *Lecture Notes in Computer Science*, pages 273–288. Springer, 2004.
20. Carlos F Lopez, Jeremy L Muhlich, John A Bachman, and Peter K Sorger. Programming biological models in python using pysb. *Molecular systems biology*, 9(1), 2013.
21. J. Lynch. A logical characterization of individual-based models. *Proceedings of Logic in Computer Science*, pages 203–217, 2008.
22. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, E. Teller, et al. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087, 1953.
23. E. Murphy, V. Danos, J. Feret, R. Harmer, and J. Krivine. Rule-based modelling and model resolution. In H. Lohdi and S. Muggleton, editors, *Elements of Computational Systems Biology*. Wiley, 2010.
24. Carl-Fredrik Tiger, Falko Krause, Gunnar Cedersund, Robert Palmér, Edda Klipp, Stefan Hohmann, Hiroaki Kitano, and Marcus Krantz. A framework for mapping, visualisation and automatic model creation of signal-transduction networks. *Molecular systems biology*, 8(1), 2012.